

# Sprint 1 Documentation

## Foundation & Access Control

**Sprint Number:** 1

**Sprint Status:** FINALIZED & READY FOR EXECUTION

**Sprint Duration:** 7 Days

**Methodology:** Scrum

---



## Sprint Goal

Establish a **secure and controlled foundation** for the application by implementing authentication, approved-user access control, database setup, and short video upload with temporary storage.

By the end of this sprint, only approved users should be able to upload short videos that are safely stored and tracked.

No AI processing is included in this sprint.

---



## User Stories Included

### US-1: User Authentication

**As a user,** I want to register and log in securely so that my identity is authenticated.

**Acceptance Criteria** - JWT-based authentication implemented - Passwords hashed securely - Protected routes enforced

---

### US-2: Approved User Access Control

**As an admin,** I want to approve users so that only authorized users can use the application.

**Acceptance Criteria** - New users are unapproved by default - Only approved users can upload videos - Admin approval mechanism exists

---

### US-3: Video Upload (Metadata + Storage)

**As an approved user,** I want to upload a short video so it can be processed later.

**Acceptance Criteria** - Max video length: 10 seconds - Video uploaded to Cloudinary - Metadata stored in MongoDB - Video status set to `uploaded` - Expiration time set to 3 hours

---

## US-4: Video Listing

**As a user,** I want to see my recent uploaded videos.

**Acceptance Criteria** - Returns last 3 videos only - Excludes expired videos - Displays current status

---

## # Technical Scope (Sprint 1 Only)

### Included

- Backend project initialization
- MongoDB Atlas connection
- Cloudinary integration
- JWT authentication
- User approval logic
- Video upload & listing APIs

### Explicitly Excluded

- Speech-to-Text
- Translation
- Text-to-Speech
- FFmpeg processing
- Background job queues



## Database Design (Sprint 1)

### Users Collection

Fields: - email (unique) - passwordHash - role (user / admin) - approved (boolean) - createdAt

Indexes: - Unique index on email

---

### Videos Collection

Fields: - userId (reference) - originalVideoUrl - status (`uploaded`) - expiresAt - createdAt

Indexes: - userId - TTL index on expiresAt

---

## API Design (Sprint 1)

### Authentication

- POST `/auth/register`
- POST `/auth/login`
- GET `/auth/me`

### Admin

- PATCH `/admin/approve/:userId`

### Videos

- POST `/videos/upload`
- GET `/videos`

---

## Data Lifecycle Rules

- Every uploaded video has an `expiresAt` timestamp
- MongoDB TTL index automatically deletes expired records
- Cloudinary assets are deleted when records expire
- Users are warned to download videos within 3 hours

---

## Definition of Done (DoD)

Sprint 1 is considered complete when:

- Users can register and log in
- Admin can approve users
- Unapproved users are blocked from uploads
- Approved users can upload videos
- Uploaded videos appear in user list
- Expired videos are automatically removed

---

## Known Limitations (Accepted)

- No AI processing
- Limited to approved users
- Short video duration only
- Manual admin approval

These limitations are **intentional design decisions**.

---

## Sprint Outcome

After Sprint 1, the system will have:

- Secure authentication
- Cost-controlled access
- Temporary media storage
- Clean database schema

This sprint lays the foundation for AI processing in subsequent sprints.

---

## Sprint 1 Conclusion

Sprint 1 establishes a **stable, secure, and scalable base** for the application.

With this foundation in place, the project is ready to move into **AI processing and media transformation** in Sprint 2.